LeaderBoard & Yesterday's Solution(/faces/candidate/leaderboarddailychallenge.xhtml?RT=DAILYCHALLENGE)

Daily Challenge **Happy Coding from necse**  **SkillRack**

Sorted List with Duplicates

Hector forms a list of integers that contains only positive integers in ascending order. The program must accept **N integers** representing the integers to be inserted or removed in the list. Initially, the list is empty. For each integer **X** among the N integers, the program must insert the integer X into the sorted list if X is positive. Otherwise, the program must remove one occurrence of the absolute value of X from the list(if exists). After each insert/remove operation, the program must print the integers in the list. If there is no integer in the list, then the program must print **EMPTY** as the output.

Boundary Condition(s):
1 <= N <= 100
-10^5 <= Each integer value <= 10^5

Input Format:
The first line contains N.
The second line contains N integer values separated by a space.

Output Format:
The first N lines, each contains integer values separated by a space or the string value EMPTY.

Example Input/Output 1:
Input:
11
10 30 10 20 10 20 10 -10 -40 20 -30

Output:
10
10 30
10 10 30
10 10 20 30
10 10 10 20 30
10 10 10 20 20 30
10 10 10 10 20 20 30
10 10 10 20 20 30
10 10 10 20 20 30
10 10 10 20 20 20 30
10 10 10 20 20 20

Explanation:
Here **N=11** and initially the list is empty.
The **1st** integer is **10** which is positive, so 10 is added to the list. Now the list becomes **10**.
The **2nd** integer is **30** which is positive, so 30 is added to the list. Now the list becomes **10 30**.
The **3rd** integer is **10** which is positive, so 10 is added to the list. Now the list becomes **10 10 30**.
The **4th** integer is **20** which is positive, so 20 is added to the list. Now the list becomes **10 10 20 30**.
The **5th** integer is **10** which is positive, so 10 is added to the list. Now the list becomes **10 10 10 20 30**.
The **6th** integer is **20** which is positive, so 20 is added to the list. Now the list becomes **10 10 10 20 20 30**.
The **7th** integer is **10** which is positive, so 10 is added to the list. Now the list becomes **10 10 10 10 20 20 30**.
The **8th** integer is **-10** which is negative, so 10 is removed from the list. Now the list becomes **10 10 10 20 20 30**.
The **9th** integer is **-40** which is negative. The list remains the same as **40** is not in the list.

The **10th** integer is **20** which is positive, so 20 is added to the list. Now the list becomes **10 10 10 20 20 20 30**. The **11th** integer is **-30** which is negative, so 30 is removed from the list. Now the list becomes **10 10 10 20 20 20**.

Example Input/Output 2:
Input:
8
-50 35 25 15 -25 -15 -35 50

Output:
EMPTY
35
25 35
15 25 35
15 35
35
EMPTY
50

Max Execution Time Limit: 50 millisecs

Ambiance

Java (12.0)

```
import java.util.*;
public class Hello {

    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int[] arr=new int[n];
        ArrayList<Integer> res=new ArrayList<>();
        for(int i=0;i<n;i++){
            arr[i]=sc.nextInt();
            if(arr[i]>=0){
                res.add(arr[i]);
            }
            else if(arr[i]<0){
                int a=Math.abs(arr[i]);
              // System.out.println(a);
               for(int j=0;j<res.size();j++){
                   if(a==res.get(j)){
                       res.remove(j);
                      break;
                       }

               }


            }
            Collections.sort(res);
            for(int k=0;k<res.size();k++){

                if(k==res.size()){
                    System.out.print(res.get(k));
                }
                else{
                System.out.print(res.get(k)+" ");
                }
            }
            if(res.size()==0){
                System.out.print("EMPTY");
            }
            System.out.println();
        }


    }
}
```

1912080@nec

Code did not pass the execution

Input:

```
8
-50 35 25 15 -25 -15 -35 50
```

Expected Output:

```
EMPTY
35
25 35
15 25 35
15 35
35
EMPTY
50
```

Your Program Output:

```
empty
35
25 35
15 25 35
15 35
35
empty
50
```

Save Run